**CS F469**

**INFORMATION RETRIEVAL**

**Domain Specific Information Retrieval System**

**Flipkart Mobile Phone Listing Search Using The Vector Space Model, with Sentiment Analysis**

**Technical Design Document**

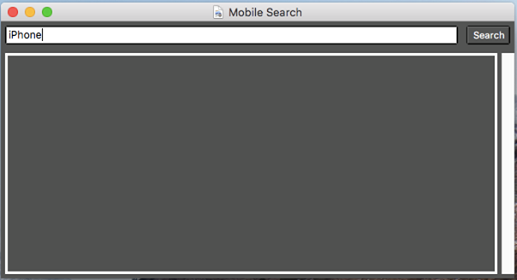
Karthik Nagaraj-2016A7PS0804H

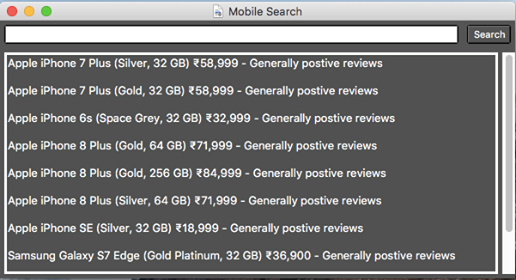
Vaibhav Balloli-2016AAPS0165H

Rohan Tammara-2016A8PS0332H

**Introduction**

The application is a python GUI-based utility that scrapes Flipkart for mobile phones, builds a vectorised dataset out of the scraped data and uses it to retrieve relevant results according to a user’s query. It also does a sentiment analysis on the scraped reviews of the phone and returns a sentiment category. It utilises the scrapy, nltk and jupyter python packages to do so.





**Overview**

**System Characteristics**

The application functions as a search engine that searches the scraped dataset to return relevant user queries, when inputted. It functions using a simple and intuitive GUI, with an entry text box, a search button to initiate the search and a text box to display results. It is meant to run locally on the user’s system. It requires no maintenance from the user’s end.

**System Context**

The setup script checks if the user’s system meets the requirements of the application. If not, it installs and/or updates the required libraries. It then begins the scraping process, which returns the phones as pickle files. On application launch, it checks if the TF-IDF vector space exists for the phones. If not, it prepares the corpus. It runs through the reviews of the phone, tokenizes them, stems them, cleans them and removes stop words from them. It then calculates the frequency matrix and the TF-IDF matrix. When the user submits a query, it tokenizes the query and calculates the scores against the TF-IDF vector space. It returns the top ten highest scored results. While returning, it passes that phone’s reviews into the sentiment analyser and returns a sentiment rating.

**Design Process**

An incremental RAD (Rapid Application Development) SDLC model was used to create the application. This model involves rapidly developing the application on incrementing GUI-based prototypes with relatively less focus on risk mitigation. As this application has next to no risk potential and is lightweight and GUI based, RAD is ideal for this case.

**Software Development Tools**

Github was used for version control.

The application was developed in a Python 3.7 environment, making use of the pip, scrapy, nltk and jupyter libraries.